Attention In Detail

- author : carrie.ywj@alibaba-inc.com

- time : 2019-06-17

Content

- First Glance of Attention
  - The History of Attention
  - “What is Attention ?”
- Attention in Details
  - Framework
  - Bahdanau Attention & Luong Attention
  - Self Attention & Multi-head Attention
  - Different Kinds of Attentions
- Applications
- Conclusion

First Glance of Attention - History

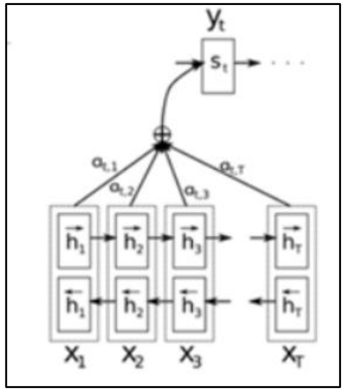
2015 - 2017
Attention

2017 - Now
Transformer

Transformer-xl

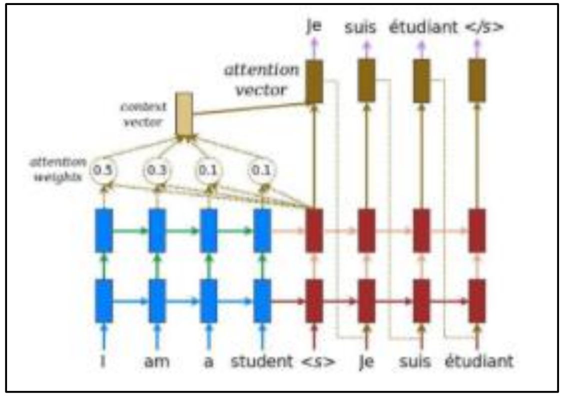
Bahdanau attention

- First Proposed 《Neural machine translation by jointly learning to align and translate》



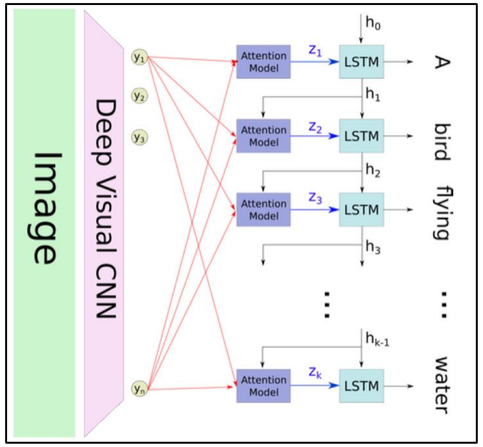
Luong attention

- Propose Several Approaches 《Effective Approaches to Attention-based Neural Machine Translation》



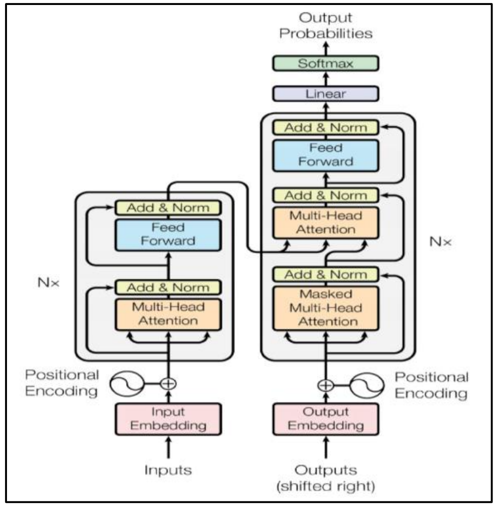
Soft/Hard attention

- Apply to Image Caption 《Show, Attend and Tell: Neural Image Caption Generation with Visual Attention》

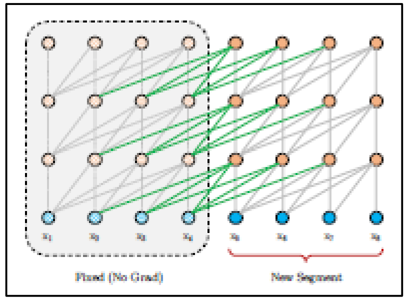


Self-attention Multihead attention

- Propose Transformer 《Attention is all you need》

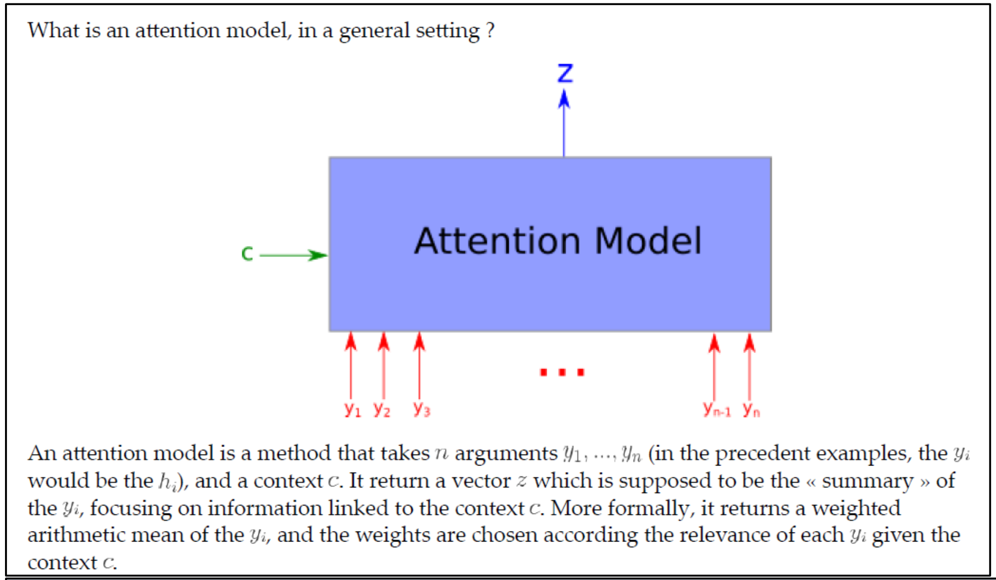


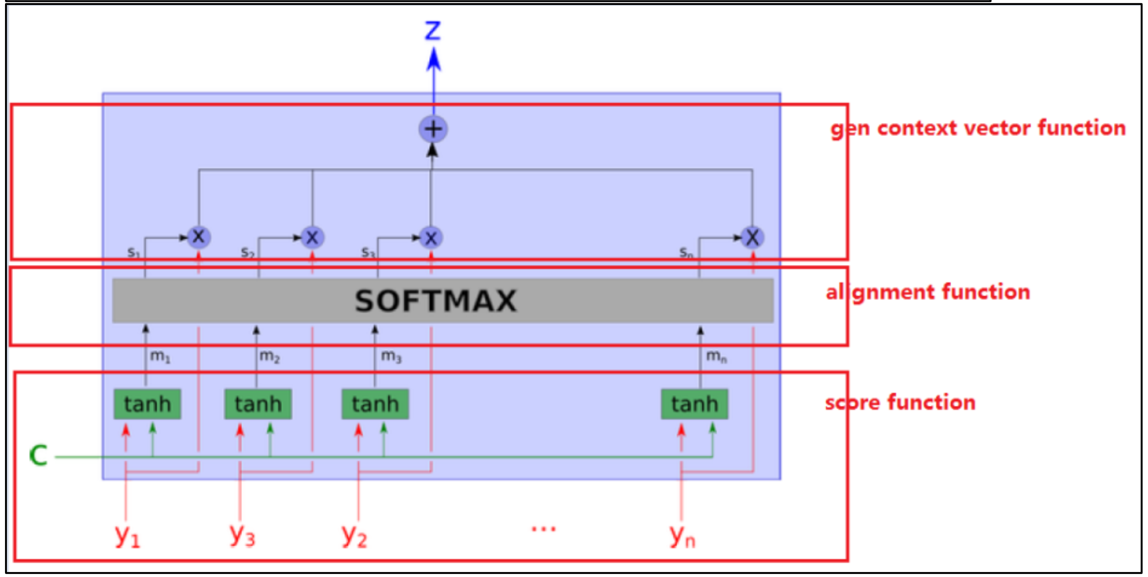
《TRANSFORMER-XL: ATTENTIVE LANGUAGE MODELS BEYOND A FIXED-LENGTH CONTEXT》



First Glance of Attention - What is Attention？

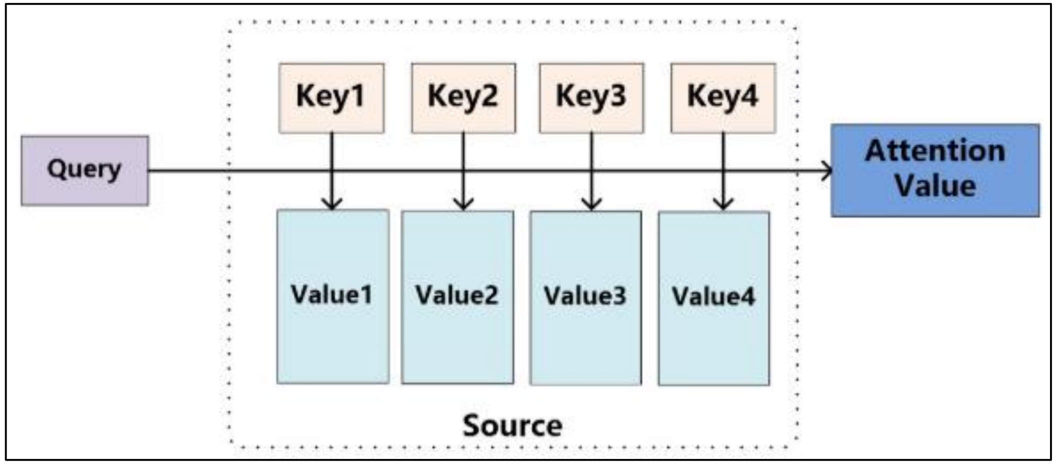
- Alignment-based (three steps)
  - Score Function
    - $e_i = a(c, y_i) = v_a^T \tanh(W_a c + U_a y_i)$
  - Alignment Function
    - $\alpha_i = softmax(e_i)$
  - Generate Context Vector Function
    - $z = \sum_i \alpha_i y_i$





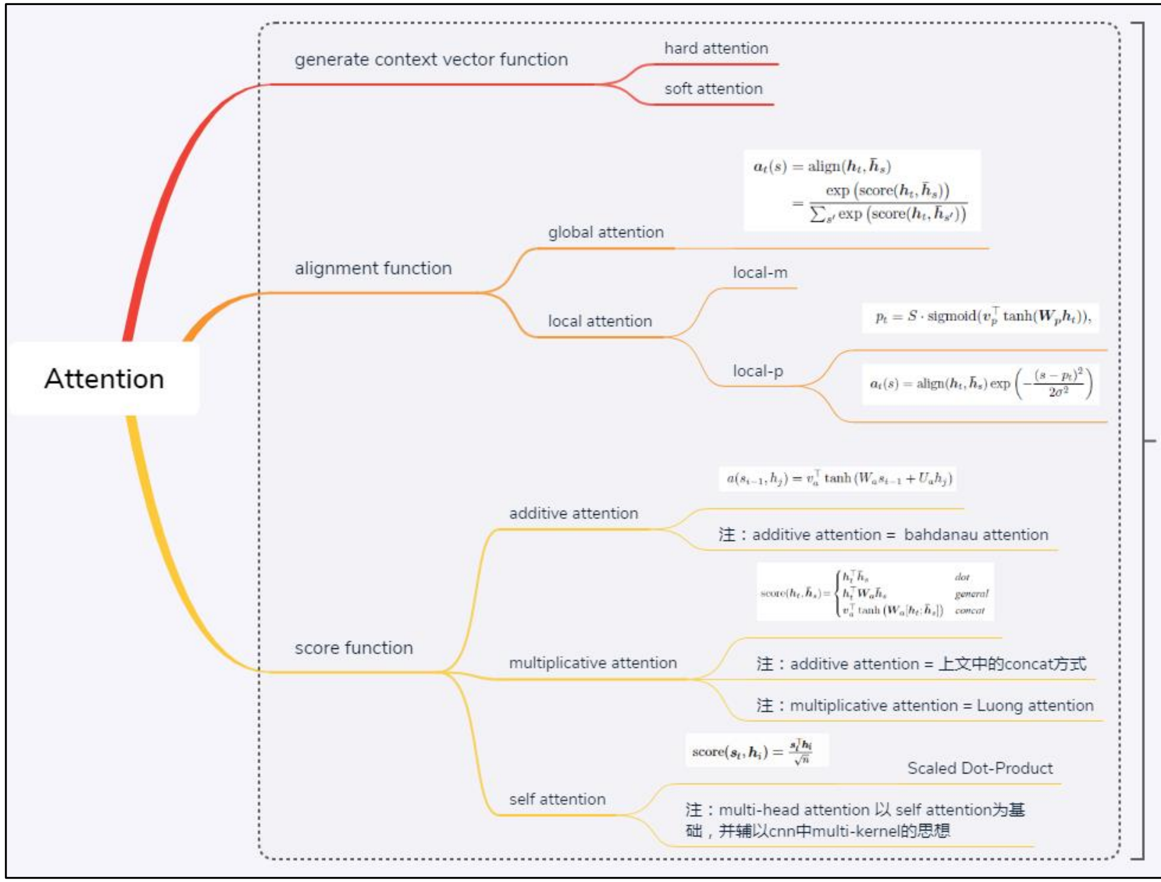
First Glance of Attention
- What is Attention？

- Memory-based (Popular in Q&A setting)
  - Address Memory (Score Function)
    - $e_i = a(q, \boxed{k_i})$
  - Normalize (Alignment Function)
    - $\alpha_i = softmax(e_i)$
  - Read Content (Generate Context Vector Function)
    - $z = \sum_i \alpha_i \boxed{v_i}$



Attention In Detail - Framework

- Perspective of “Three Steps”



Name	Alignment score function	Citation
Additive(*)	$\mathrm{score}(s_t, h_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[s_t; h_i])$	Bahdanau2015
Location-Base	$\alpha_{t,i} = \mathrm{softmax}(\mathbf{W}_a s_t)$ Note: This simplifies the softmax alignment max to only depend on the target position.	Luong2015
General	$\mathrm{score}(s_t, h_i) = s_t^\top \mathbf{W}_a h_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015
Dot-Product	$\mathrm{score}(s_t, h_i) = s_t^\top h_i$	Luong2015
Scaled Dot-Product(^)	$\mathrm{score}(s_t, h_i) = \frac{s_t^\top h_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017
Self-Attention(&)	Relating different positions of the same input sequence. Theoretically the self-attention can adopt any score functions above, but just replace the target sequence with the same input sequence.	Cheng2016
Global/Soft	Attending to the entire input state space.	Xu2015
Local/Hard	Attending to the part of input state space; i.e. a patch of the input image.	Xu2015; Luong2015

Attention In Detail - Framework

- Perspective of “Three Steps”
  - Generate Context Vector Function
    - Hard Attention
      - Stochastic “Hard”
    - Soft Attention
      - Deterministic “Soft”

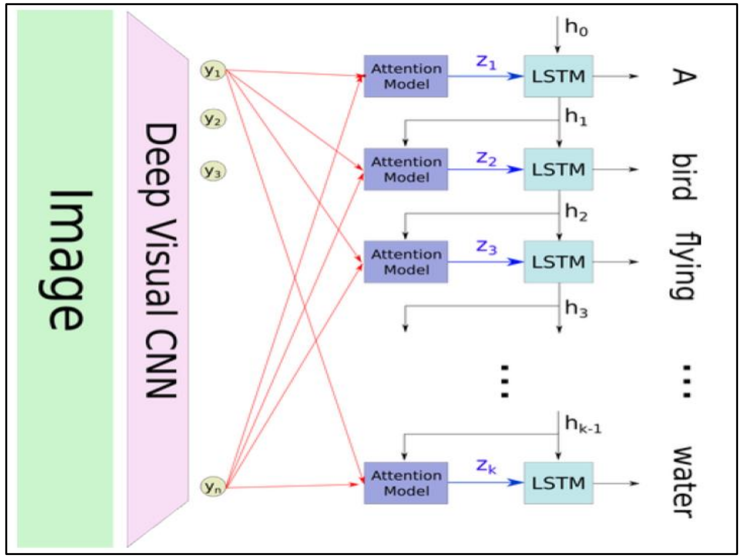
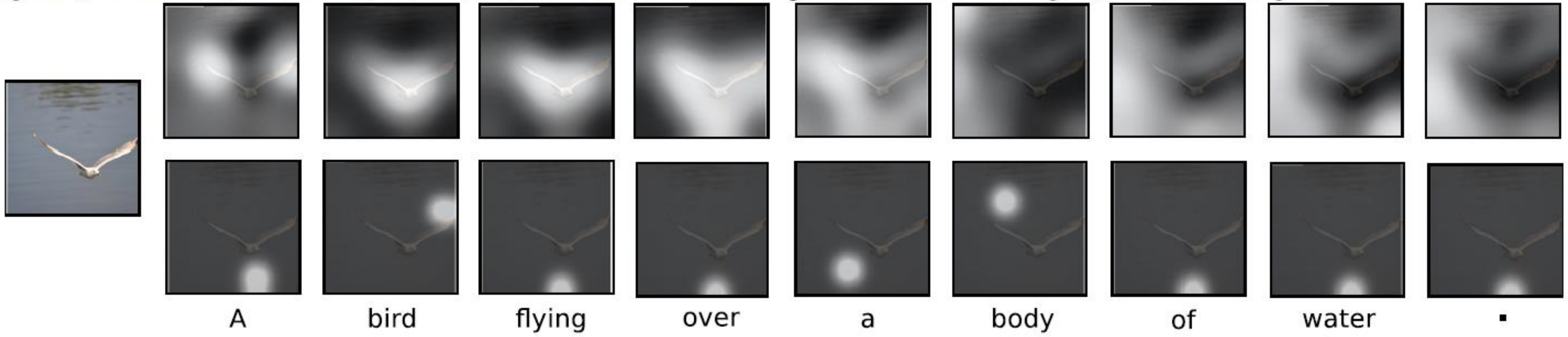


Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. “soft” (top row) vs “hard” (bottom row) attention. (Note that both models generated the same captions in this example.)



Reference：Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[C]//International conference on machine learning. 2015: 2048-2057.

https://arxiv.org/pdf/1502.03044.pdf

Attention In Detail - Framework

- Perspective of “Three Steps”
  - Alignment Function
    - Global Attention
      - Soft Attention (All the Inputs)
    - Local Attention
      - local-m
      - local-p

$$p_t = S \cdot \mathrm{sigmoid}(\boldsymbol{v}_p^\top \tanh(\boldsymbol{W}_p \boldsymbol{h}_t)), \qquad (9)$$

$$\boldsymbol{a}_t(s) = \mathrm{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \exp\left(-\frac{(s-p_t)^2}{2\sigma^2}\right) \qquad (10)$$

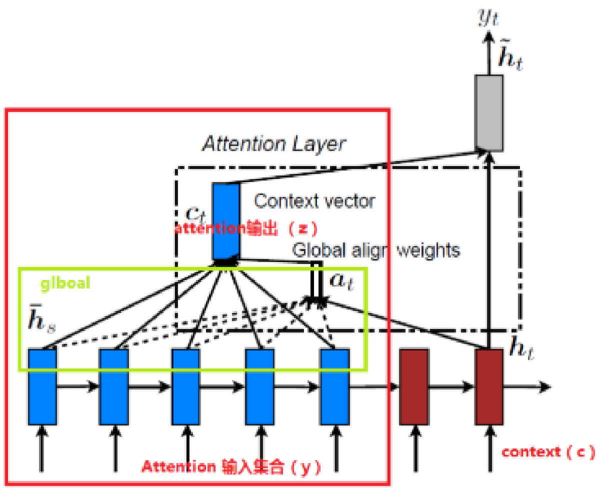


Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

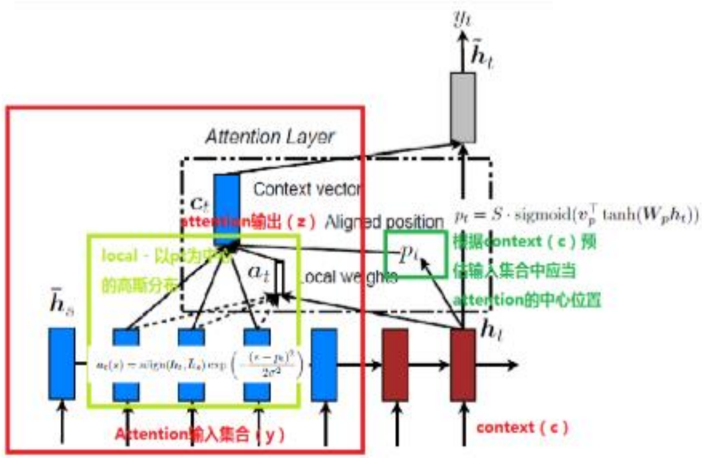


Figure 3: **Local attention model** – the model first predicts a single aligned position p_t for the current target word. A window centered around the source position p_t is then used to compute a context vector c_t, a weighted average of the source hidden states in the window. The weights a_t are inferred from the current target state h_t and those source states $\bar{h}_s$ in the window.

Reference：Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015.

https://arxiv.org/abs/1508.04025

Attention In Detail - Framework

- Perspective of “Three Steps”
  - Score Functions
    - Additive - $v_a^T \tanh(W_a s_t + U_a h_i)$
      - as “concat” in Luong,et al., 2015
      - used in Bahdanau Attention
    - Multiplicative - $s_t^T W_a h_i$
      - as “general” in Luong,et al, 2015
      - used in Luong Attention
    - Dot Product - $s_t^T h_i$
    - Scaled Dot-Product - $\frac{s_t^T h_i}{\sqrt{n}}$

Name	Alignment score function	Citation
Content-base attention	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$	Graves2014
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$	Bahdanau2015
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment to only depend on the target position.	Luong2015
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Luong2015
Scaled Dot-Product(^)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017

(*) Referred to as “concat” in Luong, et al., 2015 and as “additive attention” in Vaswani, et al., 2017.

(^) It adds a scaling factor $1/\sqrt{n}$, motivated by the concern when the input is large, the softmax function may have an extremely small gradient, hard for efficient learning.

Here are a summary of broader categories of attention mechanisms:

Name	Definition	Citation
Self-Attention(&)	Relating different positions of the same input sequence. Theoretically the self-attention can adopt any score functions above, but just replace the target sequence with the same input sequence.	Cheng2016
Global/Soft	Attending to the entire input state space.	Xu2015
Local/Hard	Attending to the part of input state space; i.e. a patch of the input image.	Xu2015; Luong2015

(&) Also, referred to as “intra-attention” in Cheng et al., 2016 and some other papers.

https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

Content

- First Glance of Attention
  - The History of Attention
  - “What is Attention ?”
- Attention in Details
  - Framework
  - Bahdanau Attention & Luong Attention
  - Self Attention & Multi-head Attention
  - Different Kinds of Attentions
- Applications
- Conclusion

Attention In Detail
- Bahdanau Attention

- Background
  - Neural Machine Translation
  - RNN Encoder-Decoder
- Bahdanau Attention
  - Learnig to Align and Translate
  - Encoder : Bi-RNN
  - Decoder : Emulates searching through a source sentence during decoding a translation
  - Yield good results on longer sentences

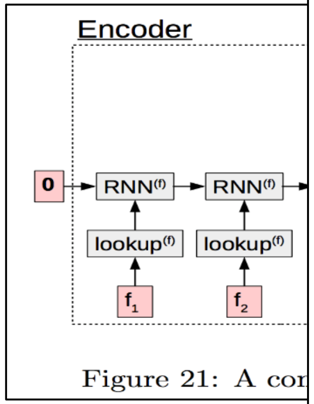


Figure 21: A co

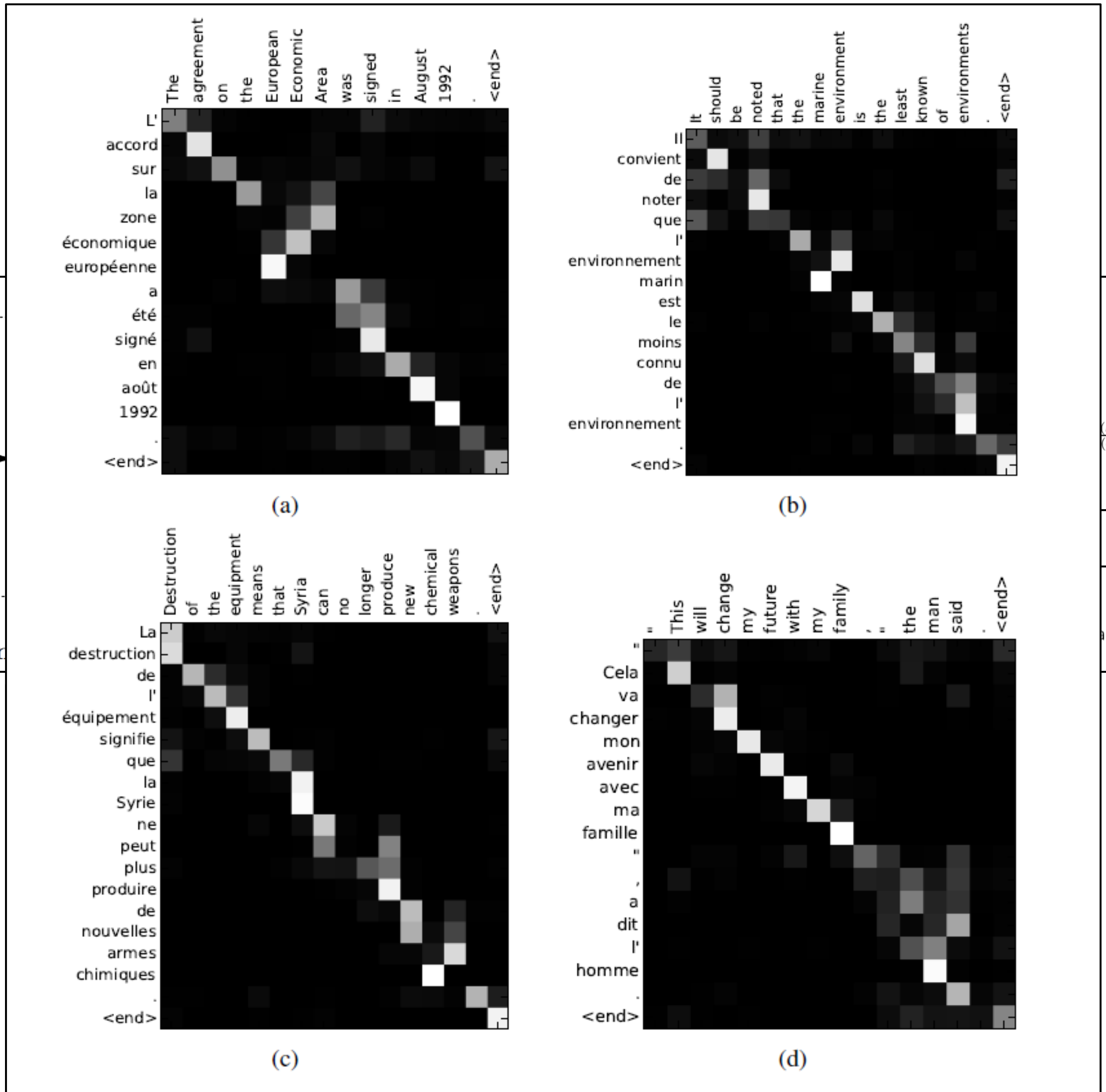


Figure 3: Four sample alignments found by RNNsearch-50. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight α_{ij} of the annotation of the j-th source word for the i-th target word (see Eq. (6)), in grayscale (0: black, 1: white). (a) an arbitrary sentence. (b–d) three randomly selected samples among the sentences without any unknown words and of length between 10 and 20 words from the test set.

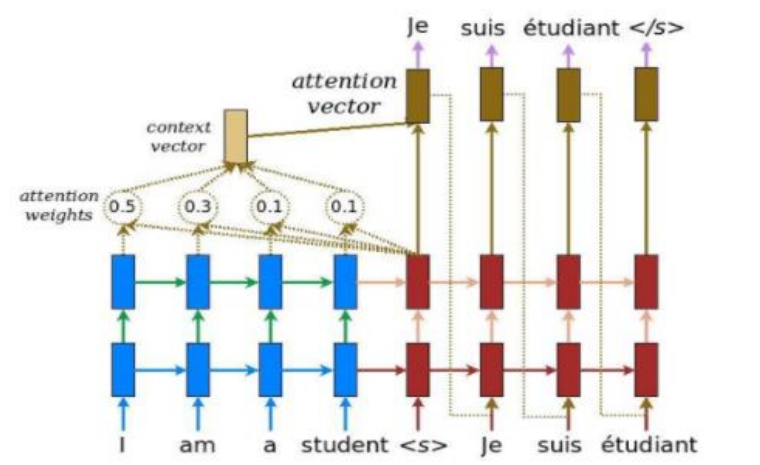
Reference：Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

Attention In Detail - Luong Attention

- Luong Attention
  - Encoder / Decoder: Stacked RNNs (4-layers)
  - Global Attention
  - Local Attention
    - Weighted Average within Window $[p_t - D, p_t + D]$
    - Monotonic alignment (**local-m**): $p_t = t$
    - Predictive alignment (**local-p**):

$$p_t = S \cdot \mathrm{sigmoid}(\boldsymbol{v}_p^\top \tanh(\boldsymbol{W}_p \boldsymbol{h}_t)), \quad (9)$$

$$\boldsymbol{a}_t(s) = \mathrm{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right) \quad (10)$$



score function	$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & dot \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & general \\ \boldsymbol{v}_a^\top \tanh(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]) & concat \end{cases}$
alignment function	$\boldsymbol{a}_t(s) = \mathrm{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \frac{\exp(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s))}{\sum_{s'} \exp(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'}))}$ or $p_t = S \cdot \mathrm{sigmoid}(\boldsymbol{v}_p^\top \tanh(\boldsymbol{W}_p \boldsymbol{h}_t)), \quad (9)$ $\boldsymbol{a}_t(s) = \mathrm{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right) \quad (10)$
context vector	$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$

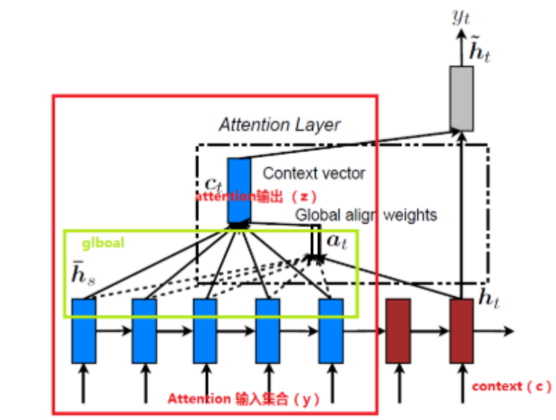


Figure 2: **Global attentional model** – at each time

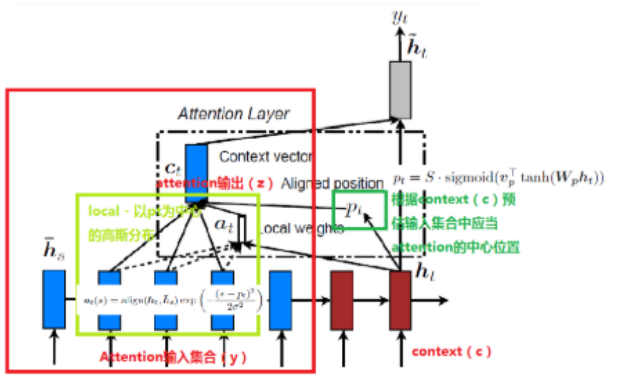


Figure 3: **Local attention model** – the model first predicts a single aligned position p_t for the current

Reference：Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015.

Attention In Detail - Luong Attention

- Result Analysis
  - Attention gives a significant boost
- Attention Architectures
  - global attention
    - dot works well
  - local attention
    - local-p (general) best

System	Ppl	BLEU	
		Before	After unk
global (location)	6.4	18.1	19.3 (+1.2)
global (dot)	6.1	18.6	20.5 (+1.9)
global (general)	6.1	17.3	19.1 (+1.8)
local-m (dot)	>7.0	x	x
local-m (general)	6.2	18.6	20.4 (+1.8)
local-p (dot)	6.6	18.0	19.6 (+1.9)
local-p (general)	**5.9**	**19**	**20.9 (+1.9)**

Table 4: **Attentional Architectures** – performances of different attentional models. We trained two local-m (dot) models; both have ppl > 7.0.

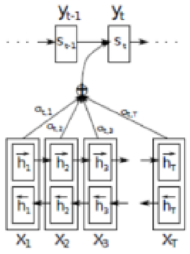
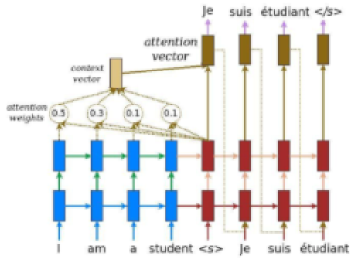
System	Ppl	BLEU
Winning WMT'14 system – *phrase-based + large LM* (Buck et al., 2014)		20.7
Existing NMT systems		
RNNsearch (Jean et al., 2015)		16.5
RNNsearch + unk replace (Jean et al., 2015)		19.0
RNNsearch + unk replace + large vocab + *ensemble* 8 models (Jean et al., 2015)		**21.6**
Our NMT systems		
Base	10.6	11.3
Base + reverse	9.9	12.6 (+*1.3*)
Base + reverse + dropout	8.1	14.0 (+*1.4*)
Base + reverse + dropout + global attention (*location*)	7.3	16.8 (+*2.8*)
Base + reverse + dropout + global attention (*location*) + feed input	6.4	18.1 (+*1.3*)
Base + reverse + dropout + local-p attention (*general*) + feed input	5.9	19.0 (+*0.9*)
Base + reverse + dropout + local-p attention (*general*) + feed input + unk replace		20.9 (+*1.9*)
Ensemble 8 models + unk replace		**23.0** (+*2.1*)

Table 1: **WMT'14 English-German results** – shown are the perplexities (ppl) and the *tokenized* BLEU scores of various systems on newstest2014. We highlight the **best** system in bold and give *progressive* improvements in italic between consecutive systems. *local-p* referes to the local attention with predictive alignments. We indicate for each attention model the alignment score function used in pararentheses.

Reference：Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015.

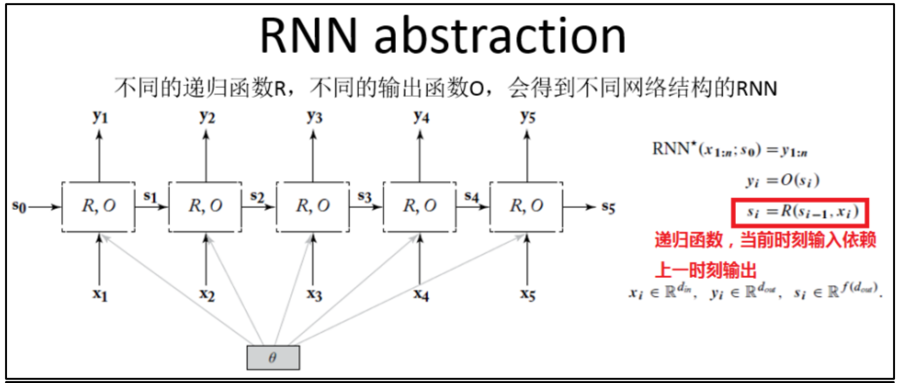
Attention In Detail -Compare

- Same
  - Soft Attention Used in Decoder
- Different
  - Context Setting
  - Input Feeding
  - Encoder & Decoder RNN

	Bahdanau（2015）	Luong（2015）	差异
别名	additive attention Bahdanau attention	multiplicative attention Luong attention	Bahdanau是最经典的attention结构，Luong则在基础上尝试不同score-alignment function
框架图			encoder网络：前者使用双向RNN，后者使用单向多层RNN。 decoder网络：后者使用多层RNN，且增加input feeding。 context设置：前者使用上一步decoder的隐状态，后者使用当前term在顶层RNN的因状态。
score function	$e_{ij} = a(s_{i-1}, h_j)$ $a(s_{i-1}, h_j) = v_a^\top \tanh(W_a s_{i-1} + U_a h_j)$	$\mathrm{score}(h_t, \bar{h}_s) = \begin{cases} h_t^\top \bar{h}_s & dot \\ h_t^\top W_a \bar{h}_s & general \\ v_a^\top \tanh(W_a[h_t; \bar{h}_s]) & concat \end{cases}$	前者score function 与 后者的concat是一样的。 后者尝试了更多的score fucntion。
alignment function	$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$,	$a_t(s) = \mathrm{align}(h_t, \bar{h}_s) = \frac{\exp(\mathrm{score}(h_t, \bar{h}_s))}{\sum_{s'} \exp(\mathrm{score}(h_t, \bar{h}_{s'}))}$ or $p_t = S \cdot \mathrm{sigmoid}(v_p^\top \tanh(W_p h_t)), \quad (9)$ $a_t(s) = \mathrm{align}(h_t, \bar{h}_s) \exp\left(-\frac{(s-p_t)^2}{2\sigma^2}\right) \quad (10)$	基础版本都是softmax。 在Luong中尝试了多种对齐方式。 local-p加了一个预估流程；
context vector	$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$	$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$	same（都是soft的方式） local attention会限制 j 范围在[pt-D, pt+D]窗口内

Attention In Detail
-Self Attention

- Why Self-Attention ?
  - As Feature Extractor
    - RNN
      - Long Dependency : A little tricky
      - Sequence : Can't handle hierarchical information
      - Recurrent : No parallelize
    - CNN
      - N-gram detector : Local dependency
      - Hierarchical Receptive Field : Logarithmic path length
      - Parallelize within One-Layer
    - Self Attention
      - Constant Path Length
      - Variable-sized Perceptive Field
      - Parallelize Per Layer



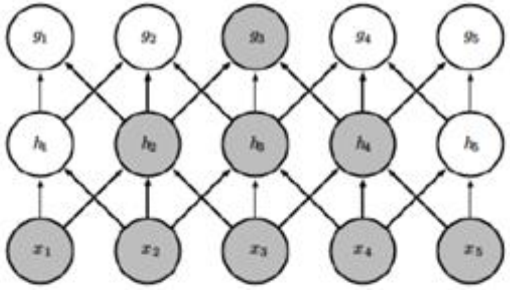
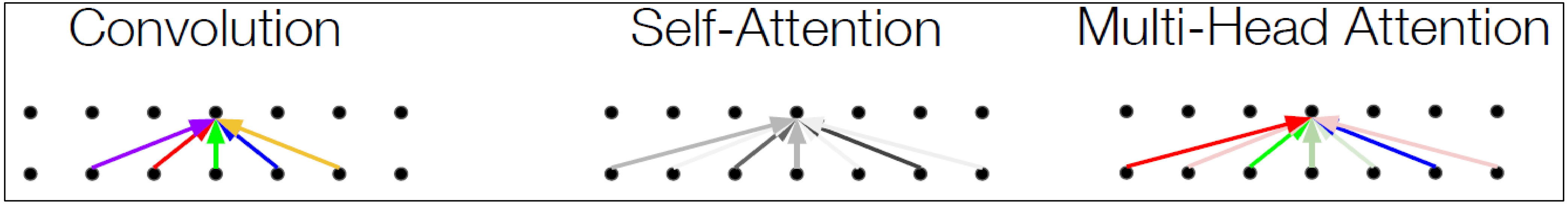


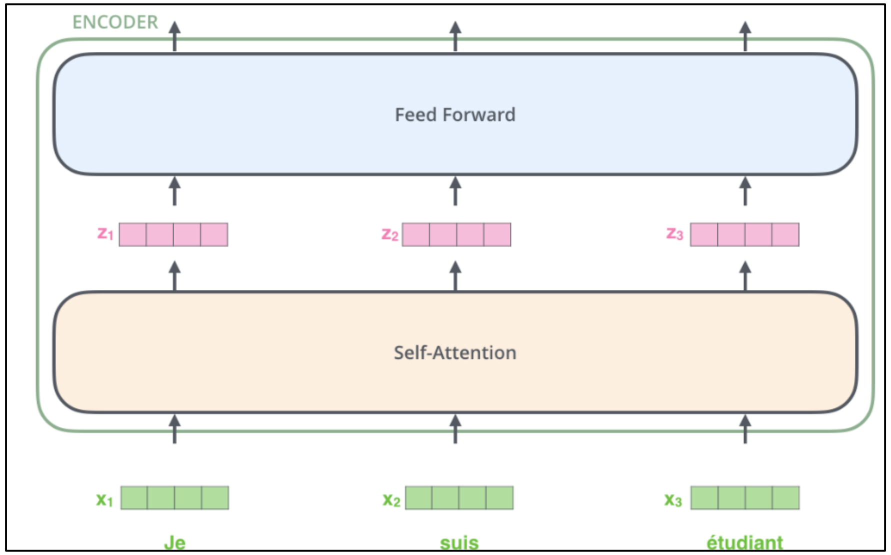
Figure 9.4: The receptive field of the units in the deeper layers of a convolutional network is larger than the receptive field of the units in the shallow layers. This effect increases if the network includes architectural features like strided convolution (figure 9.12) or pooling (section 9.3). This means that even though *direct* connections in a convolutional net are very sparse, units in the deeper layers can be *indirectly* connected to all or most of the input image.

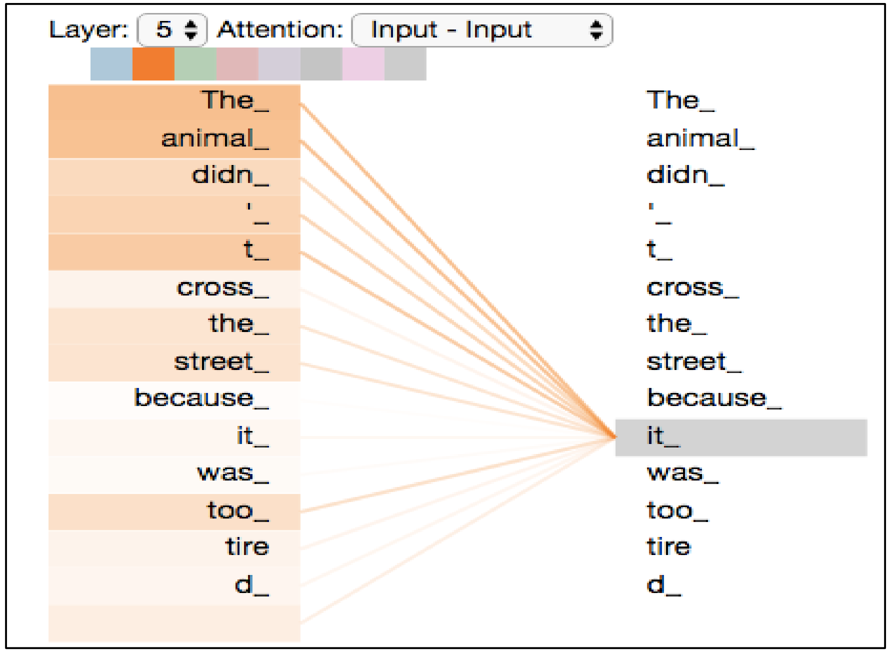


Reference :https://nlp.stanford.edu/seminar/details/lkaiser.pdf

Attention In Detail
-Self Attention

- A High-Level Look
  - Input & Output of Self-Attention
  - Look at other positions in the input sequence
  - Understanding of other relevant words into the one

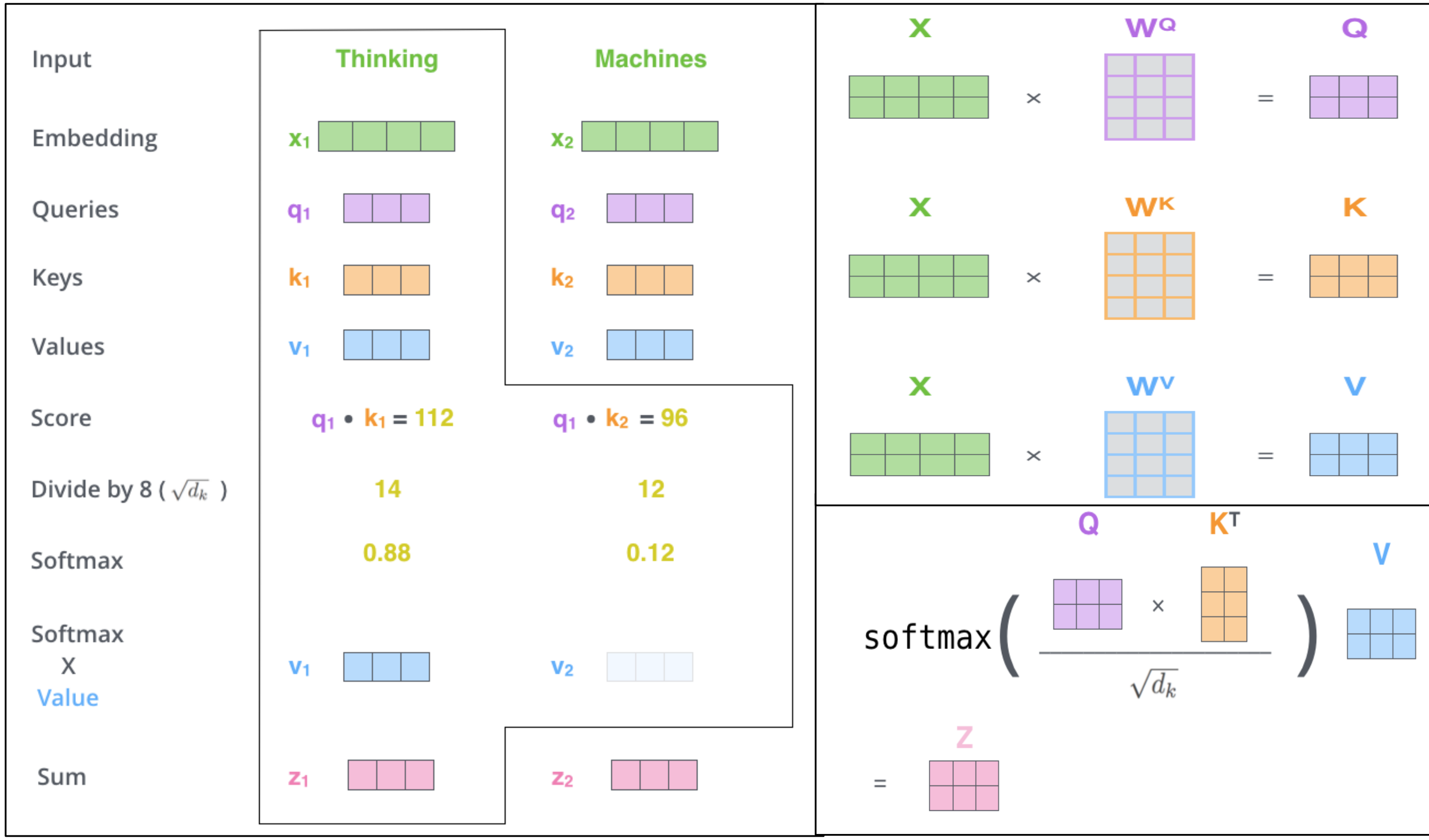




Reference :https://jalammar.github.io/illustrated-transformer/

Attention In Detail
-Self Attention

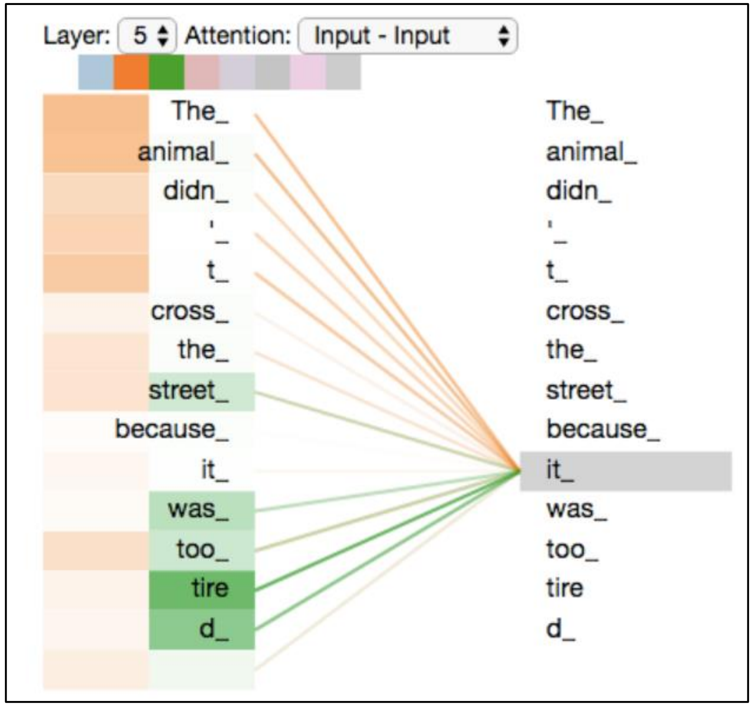
- Self Attention In Detail
  - QKV
    - Flexible
  - Scaled Dot Product
    - Leads more stable gradients
  - Advantage
    - Constant Path Length
    - Variable-sized Perceptive Field
    - Parallelize Per Layer

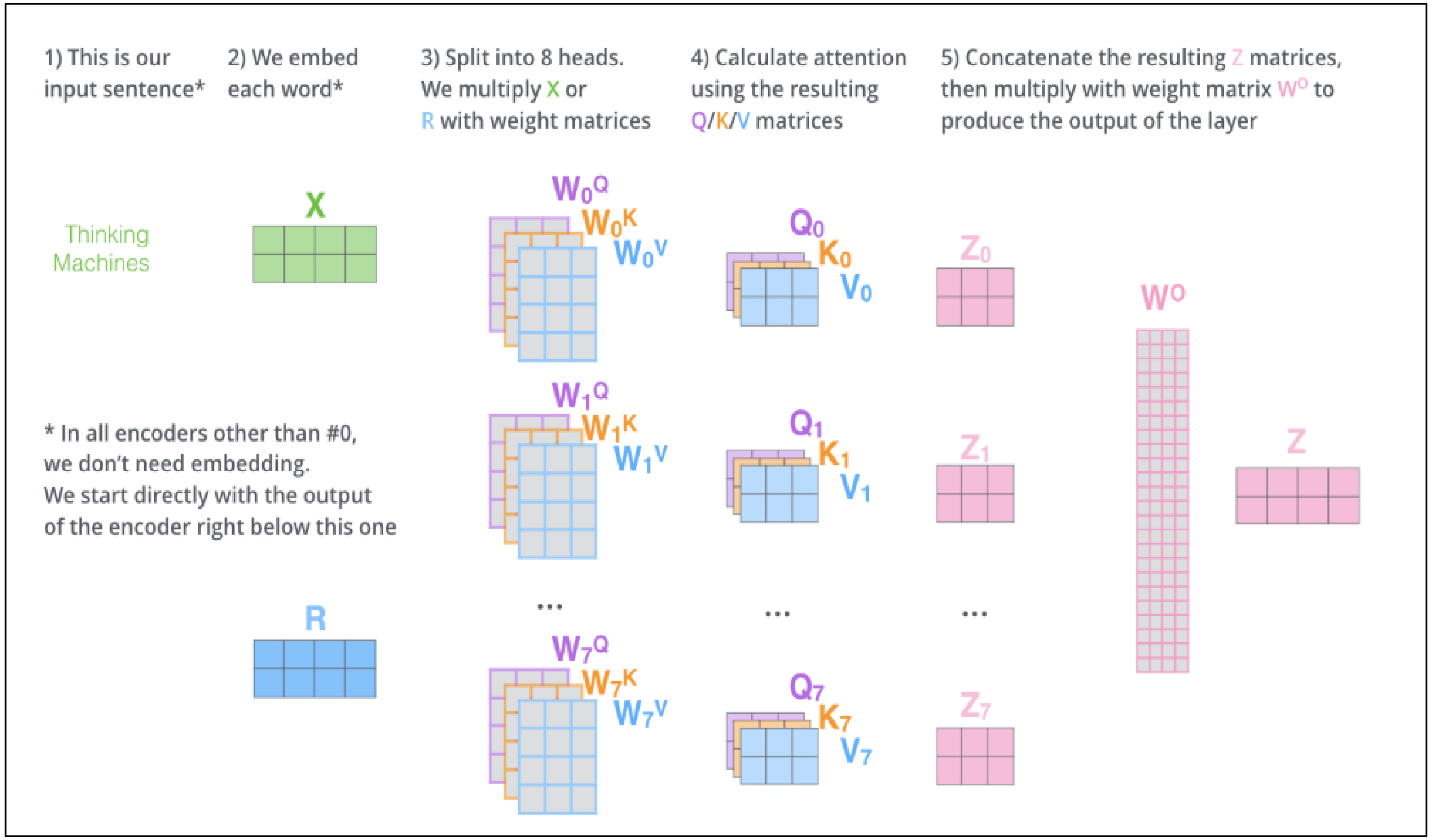


Reference :https://jalammar.github.io/illustrated-transformer/

Attention In Detail
-Multi-Head Attention

- Multi-Head Attention
  - Pretty like Multi-Kernel
  - Representation Subspace

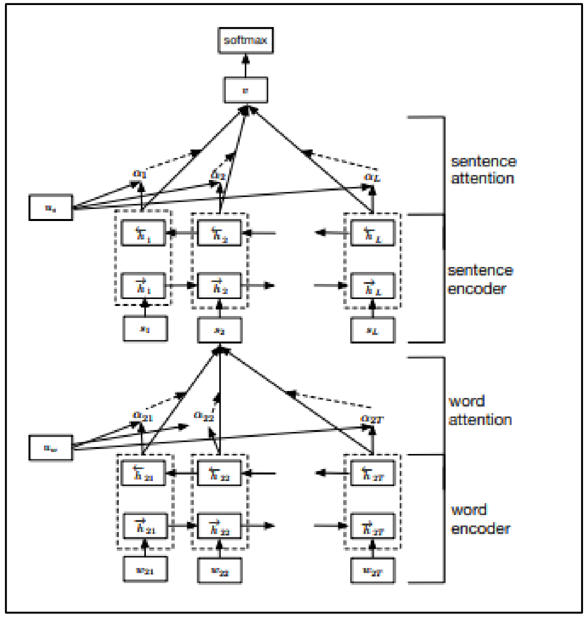


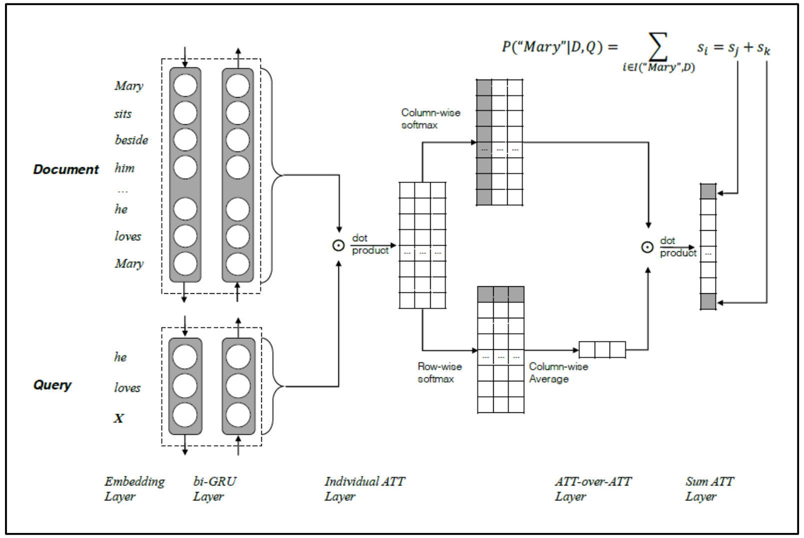


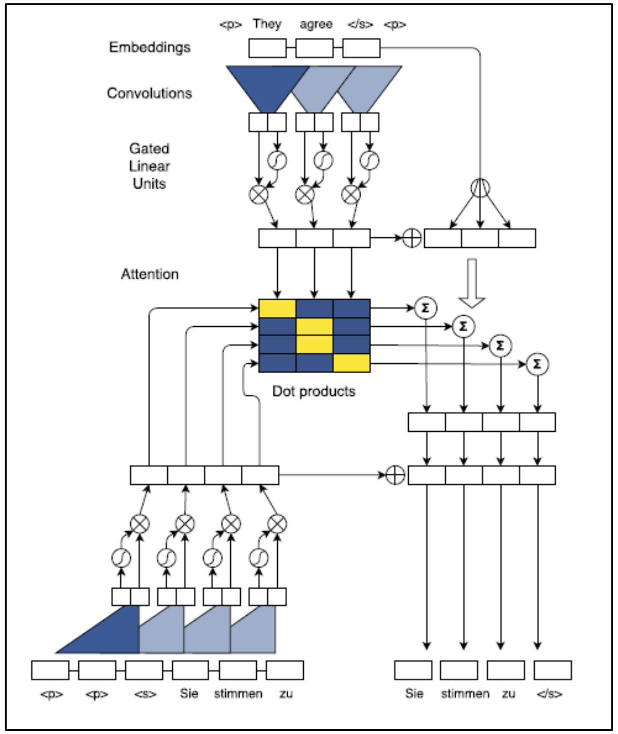
Reference :https://jalammar.github.io/illustrated-transformer/

Attention In Detail
-Different Kinds of Attentions

- Hierarchical Attention | Attention-over-Attention | Multi-Step Attention







Reference :
Yang Z, Yang D, Dyer C, et al. Hierarchical attention networks for document classification[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 1480-1489.
Cui Y, Chen Z, Wei S, et al. Attention-over-attention neural networks for reading comprehension[J]. arXiv preprint arXiv:1607.04423, 2016.
Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 1243-1252.